

Problem A. 自动装配机自动装配自动机

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

agKc 是一名《炉石传说》玩家，而其中有一种名叫"酒馆战棋"的模式。在该模式中，每名玩家需要选择不同的随从组成自己的战队来互相战斗，由于战队中能放下的随从数量有限，因此合理安排不同的随从就显得至关重要。



我们对其中一种叫作"星元自动机"的流派做出简化，其核心随从为：

- 星元自动机：具有 a 点攻击力
- 自动装配机：具有 1 点攻击力，并能够使所有的"星元自动机"获得 b 点攻击力

作为一个不怎么接触该模式的玩家，*agKc* 想要问问你，如果你的战队中最多只能容纳 n 位随从，怎么安排"星元自动机"与"自动装配机"的数量才能使战队的总攻击力最高。

Input

每组样例含有多组测试点。

第一行一个正整数 T ($1 \leq T \leq 10^5$) 表示测试点数量。每个测试点输入格式如下

一行分别输入三个整数 n, a, b ($1 \leq n \leq 10^5$, $0 \leq a, b \leq 10^8$)

Output

共 T 行

每行一个整数，表示战队最大总攻击力。

Examples

standard input	standard output
3	1
1 1 1	47
7 2 3	30
5 6 1	
3	23
5 3 2	35
5 5 3	4
1 4 1	

Problem B. 最短路图

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

坎格鲁斯普雷最近学了最短路有关的算法，于是他给袋鼠将军出了这样一个问题：

给定一张有 n 个节点的图，图中初始没有任何边，然后给你 m 条有权值的没有被指定连接的节点的（你可以任意决定它所连接的节点）无向边，你需要将这 m 条无向边全部加进图内（每条边只能被使用一次），不允许出现自环（但允许出现重边），使得 1 号节点到其他所有节点的最短路恰好为一个给定的数组 dis 。

袋鼠将军显然并不是图论高手，他并不会这道问题，于是他求助了你。当然，袋鼠将军也不是一个贪心的人，你并不需要构造问题的解决方案，你只需要判断问题是否有解即可。

Input

本题包含多组测试数据

输入第一行一个整数 T ，表示测试数据组数。（ $1 \leq T \leq 500$ ）

对于每组测试数据，第一行两个整数 n, m 。（ $1 \leq n, m \leq 500$ ）

第二行 m 个整数 w_i ，表示第 i 条边的权值。（ $0 \leq w_i \leq 10^9$ ）

第三行 n 个整数 dis_i ，表示题目给定的数组， dis_i 表示 1 号节点到 i 号节点的最短路长度。（ $0 \leq dis_i \leq 10^9$ ）

数据保证 $\sum n, \sum m$ 均不大于 500。

Output

对于每组测试数据，若有解，输出一行一个字符串 YES；若无解，输出一行一个字符串 NO。评测时不区分大小写，也就是说，YeS, yes, nO 都会被认为是正确的答案。

Example

standard input	standard output
2	YES
6 8	NO
1 1 2 2 3 4 4 4	
0 5 9 3 8 7	
5 8	
1 1 2 2 3 4 4 4	
0 5 9 3 8	

Note

对于样例 #1:

对于第一组测试数据，一种合法的方案是：

边1连接5号节点和6号节点；

边2连接5号节点和3号节点；

边3连接2号节点和4号节点；

边4连接2号节点和6号节点；

边5连接1号节点和3号节点；

边6连接3号节点和1号节点；

边7连接5号节点和6号节点;

边8连接4号节点和2号节点。

对于第二组测试数据，可以证明，无论以何种方案安排边，都无法满足题目的要求。

Problem C. 总辖之愿

Input file: standard input
Output file: standard output
Time limit: 1.5 seconds
Memory limit: 512 megabytes

璃寒和曜灵是一对好朋友。

某一天，璃寒告诉曜灵，一种名为"挖矿"的方法非常赚钱，于是曜灵就想要前往另一个星球去挖矿，虽然她的理解完全错误，但是她真的找到了一个可以视为具有 $n \times m$ 的网格的矩形的富含矿物的星球，她在这颗星球上建造了中心点互不重合的 k 座矿场，一座编号为 i 的矿场可以被它的中心点坐标 (x_i, y_i) 和延伸距离 dis_i 确定，具体地，一个网格 (x_{now}, y_{now}) 在这个矿场，当且仅当 x_{now} 与 y_{now} 不超过该星球的范围且同时满足 $|x_i - x_{now}| \leq dis_i$ 和 $|y_i - y_{now}| \leq dis_i$ 。

建造完这些矿场后，曜灵还需要运输线确保这些矿物能够送回地球进行售卖，真正达到赚钱的目的，于是她还需要确定一个中心点在 (x_{ans}, y_{ans}) 、支撑距离 dis_{ans} 且所有占据的格子不在矿场中的发射台以方便火箭来往，一个网格 (x_{lanuch}, y_{lanuch}) 被发射台占据，当且仅当 x_{lanuch} 与 y_{lanuch} 不超过该星球的范围且 $|x_{ans} - x_{lanuch}| + |y_{ans} - y_{lanuch}| \leq dis_{ans}$ 。

由于曜灵非常聪明，所以你只需要输出符合条件的发射台的最大支撑距离 dis_{ans} 即可，曜灵会自己找位置建的，如果曜灵没那么聪明，让你找不到任何一个可以建造的位置，请输出 '-1' (不包含单引号)。

需要注意的是，矿场和发射台占据的区域可以超过星球，但是中心点一定需要在星球中。

Input

第一行，三个整数 $n(1 \leq n \leq 500), m(1 \leq m \leq 500), k(1 \leq k \leq 500)$ ，表示矩形的大小和矿场个数。

第 2 行到第 $k + 1$ 行每行三个整数，其中第 $i + 1$ 行的整数为 $x_i(1 \leq x_i \leq n), y_i(1 \leq y_i \leq m), dis_i(0 \leq dis_i \leq 500)$ ，表示第 i 座矿场的中心点和延伸距离。

Output

共一行，一个整数 dis_{ans} ，表示符合条件的发射台的最大支撑距离。

Examples

standard input	standard output
5 6 2 2 2 2 5 6 1	1
5 5 1 3 3 2	-1

Note

样例 1 解释：

矿场	矿场	矿场	矿场	发射台	<u>1</u>
矿场	<u>2</u>	矿场	矿场	发射台	<u>1</u>
矿场	矿场	矿场	矿场		发射台
矿场	矿场	矿场	矿场	矿场	矿场
				矿场	<u>1</u>

带下划线的数字为矿场的中心点，矿场格已被标出，此时中心点为 (1,6) 或 (2,6)，支撑距离为 1 的发射台是支撑距离最大的选择地点。

样例 2 解释：

由于矿场的范围占据了整颗星球，所以没有任何一个位置建造发射台，输出 '-1'。

Problem D. 恋恋的心跳大冒险 III

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 512 megabytes

agKc仍在等待着这场大冒险的结局

距离前两次大冒险已过去一段时日，但一场异变又将一切带回了原地，宝石再次被分散到了幻想乡的各处，理所当然，恋恋发起了一场新的大冒险。



受异变的影响，幻想乡的地图可视为一棵 N 个节点且以 1 为根的有根树，每个节点上都有一种宝石，每种宝石都有其对应的能量级别。

一场大冒险有两位参与者，分为多次游戏，每次游戏中，双方轮流进行如下操作：

选取一个节点作为起点，收集以该点为根的子树（包括起点）上的宝石，并将所能形成的最长连续能量级别的长度作为分数。

参与者每次操作都必须获得比对方上次操作高的分数，如果无法进行操作，则该次游戏结束。所有的节点在整场大冒险中只能被选为一次起点，但节点上的宝石能被多次收集。

你与 agKc 正在观看幻想乡中发生的每场大冒险，而 agKc 却无意识地向你提出了 Q 次询问：如果一场大冒险只能选择编号 $[L, R]$ 的点作为起点，那么这次大冒险最少和最多进行多少次游戏呢？

凭借丰富的算法知识，你很快给出了回答。

Input

第一行两个正整数 N, Q ($1 \leq N, Q \leq 1 \times 10^5$), 分别表示节点数和询问数。

第二行 N 个正整数 $x_1, x_2, \dots, x_N$ ($1 \leq x_1, x_2, \dots, x_N \leq N$), 表示 i 号节点上的宝石能量级别为 x_i 。

接下来 $N - 1$ 行，每行两个正整数 u, v ($1 \leq u, v \leq N$), 表示存在一条从节点 u 到节点 v 的边。

最后 Q 行，每行两个正整数 L, R ($1 \leq L \leq R \leq N$), 表示询问中的 $[L, R]$ 。

Output

共 Q 行

第 i 行两个整数，分别表示第 i 次询问中的大冒险进行的最少的游戏数与最多的游戏数。

Example

standard input	standard output
4 4	2 4
1 2 2 3	1 2
1 2	1 2
2 3	1 3
2 4	
1 4	
2 3	
1 2	
1 3	

Note

样例解释

选择 1 号节点作为起点，可以收集到能量级别为 1,2,2,3 的宝石，此时得分为 3

选择 2 号节点作为起点，可以收集到能量级别为 2,2,3 的宝石，此时得分为 2

选择 3 号节点作为起点，可以收集到能量级别为 2 的宝石，此时得分为 1

选择 4 号节点作为起点，可以收集到能量级别为 3 的宝石，此时得分为 1

对于询问 [1,4]，最少的游戏数为 2，最多的游戏数为 4

对于询问 [2,3]，最少的游戏数为 1，最多的游戏数为 2

对于询问 [1,2]，最少的游戏数为 1，最多的游戏数为 2

对于询问 [1,3]，最少的游戏数为 1，最多的游戏数为 3

可以证明没有更多或更少的游戏次数

Problem E. 怯战蜥蜴 VI

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

经过数与袋鼠将军的鏖战，坎格鲁斯普雷终于打到了袋鼠将军的基地！袋鼠将军的基地是一棵由 n 个节点构成的树，其中袋鼠将军位于 x 号节点。

坎格鲁斯普雷决定从这棵树的某个叶子（度数为 1 的点）入侵袋鼠将军的基地，然后再通过最短路径来到 x 号节点与袋鼠将军进行单挑。但不幸的是，袋鼠将军的爪牙会在路上阻拦他，第 i 个节点驻守有袋鼠将军的一名爪牙的概率为 p_i ，一个节点最多驻守一名爪牙，且爪牙不会在节点间相互移动。

幸运的是，坎格鲁斯普雷是从袋鼠大学网安专业毕业的袋鼠，在入侵袋鼠将军的基地之前（此时坎格鲁斯普雷还没进入这棵树），他会黑进袋鼠将军基地的服务器，从而知晓哪些节点驻守了坎格鲁斯普雷的爪牙，从而选择一条遭遇爪牙数量最少的路径来到 x 号节点。

坎格鲁斯普雷要对这次作战计划进行详细的评估，以衡量风险与价值。现在，他想问问你，分别在 $x = 1, 2, 3, \dots, n$ 的情况下，在坎格鲁斯普雷黑进袋鼠将军基地的服务器之前，坎格鲁斯普雷在本次作战计划中遭遇的爪牙数量的期望是多少（在 x 号节点遭遇到的爪牙也计入答案）？为了避免精度误差，你只需要输出答案对 998244353 取模后的结果。

Input

输入第一行一个整数 n 。（ $2 \leq n \leq 3000$ ）

在本题中概率的输入方式为：我们将分别给出 p_i 的分子 a_i 和分母 b_i ，则 $p_i = \frac{a_i}{b_i}$ 。

第二行 n 个整数 a_i ，表示 p_i 的分子。（ $0 \leq a_i \leq 998244352$ ）

第三行 n 个整数 b_i ，表示 p_i 的分母。（ $1 \leq b_i \leq 998244352$ ）

数据保证有 $a_i \leq b_i$ 。

之后 $(n-1)$ 行，每行两个整数 u_i, v_i ，表示树内存在一条连接节点 u_i 和节点 v_i 的边。（ $1 \leq u_i, v_i \leq n$ ）

Output

输出一行 n 个整数表示答案，整数之间用空格隔开，第 i 个整数表示当 $x = i$ 情况下的答案，答案需对 998244353 进行取模。

Examples

standard input	standard output
3 1 1 2 2 3 3 1 2 1 3	388206138 332748118 665496236
6 1 2 3 4 5 6 2 3 4 5 6 7 1 2 1 3 2 4 2 5 4 6	样例2的输出见下发文件或Note部分

Note

对于**样例 #1**:

以 $x = 1$ 为例:

当 1,2,3 号节点上均驻守有爪牙时, 概率为 $\frac{1}{2} \times \frac{1}{3} \times \frac{2}{3} = \frac{1}{9}$, 这时坎格鲁斯普雷会遭遇 2 名爪牙。

当 1,2 号节点上驻守有爪牙, 但 3 号节点上没有驻守爪牙时, 概率为 $\frac{1}{2} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{18}$, 这时坎格鲁斯普雷会遭遇 1 名爪牙。

当 1,3 号节点上驻守有爪牙, 但 2 号节点上没有驻守爪牙时, 概率为 $\frac{1}{2} \times \frac{2}{3} \times \frac{2}{3} = \frac{2}{9}$, 这时坎格鲁斯普雷会遭遇 1 名爪牙。

当 1 号节点上驻守有爪牙, 但 2,3 号节点上没有驻守爪牙时, 概率为 $\frac{1}{2} \times \frac{2}{3} \times \frac{1}{3} = \frac{1}{9}$, 这时坎格鲁斯普雷会遭遇 1 名爪牙。

当 2,3 号节点上驻守有爪牙, 但 1 号节点上没有驻守爪牙时, 概率为 $\frac{1}{2} \times \frac{1}{3} \times \frac{2}{3} = \frac{1}{9}$, 这时坎格鲁斯普雷会遭遇 1 名爪牙。

其他情况下, 坎格鲁斯普雷均不会遭遇到袋鼠将军的爪牙。

故此时答案为 $\frac{1}{9} \times 2 + \frac{1}{18} \times 1 + \frac{2}{9} \times 1 + \frac{1}{9} \times 1 + \frac{1}{9} \times 1 = \frac{13}{18}$ 。 $\frac{13}{18}$ 在模 998244353 意义下为 388206138。

对于**样例 #2**:

因 pdf 渲染问题, 此样例的输出在此展示:

130722476 623902722 249561089 286401060 831870295 142606337

为什么这个怯战蜥蜴系列的题目中题面没有怯战蜥蜴呢?

Problem F. Major

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

奥斯汀 *Major* 开幕了！*RalZarek* 为了更加享受比赛，购买了观众通行证并加入了竞猜的行列。第一轮竞猜的顺利通过让他以为竞猜是件简单的事情，仅需跟随大主播的选择，将队伍按照实力排序就能轻松通过。然而第二轮的爆冷让他认清了现实。

为了不再之后的竞猜中重蹈覆辙，*RalZarek* 在第三轮中将自己完成竞猜。根据他的观赛经验，他已经对接下来要进行瑞士轮的队伍水平进行了自己的评判，给出了胜负关系表，但他仍需要研究清楚瑞士轮是个怎么样的赛制，于是他找到了官方文档，并仔细研读起来。

官方文档中对瑞士轮的解释如下：

- 瑞士轮中有且仅有16支队伍，相同胜负场记录的队伍之间将进行比赛。
- 3胜或3负将结束比赛。
- 每一轮中每支没有结束比赛的队伍将进行一场比赛且不会发生重赛，不会发生重赛是指两队在某一轮进行过对战后，在之后的轮次中即使属于相同的胜负场记录组也不会被安排进行再次比赛。
- 每一轮中对战安排将以种子排名决定。
 - 在第一轮中（0胜0负），种子排名 1 – 8 的队伍将分别与种子排名 9 – 16 的队伍对战。（1 对 9, 2 对 10, 依此类推）
 - 在第二轮（1胜0负，0胜1负）和第三轮（2胜0负，1胜1负，0胜2负），相同战绩的队伍之间将会进行比赛，种子排名最高的队伍将优先选择种子排名最低且不会发生重赛的队伍。
 - 在第四轮（2胜1负，1胜2负）和第五轮（2胜2负），每个胜负场组中将有六只队伍，将其按种子排名重新排序后从高到低标记为1至6队，按照下方的表格从上往下的优先级中第一个满足不会发生重赛条件的对阵情况比赛。

	优先级	对阵情况
	1	1v6 2v5 3v4
	2	1v6 2v4 3v5
	3	1v5 2v6 3v4
	4	1v5 2v4 3v6
	5	1v4 2v6 3v5
	6	1v4 2v5 3v6
第四和第五轮对阵情况	7	1v6 2v3 4v5
	8	1v5 2v3 4v6
	9	1v3 2v6 4v5
	10	1v3 2v5 4v6
	11	1v4 2v3 5v6
	12	1v3 2v4 5v6
	13	1v2 3v6 4v5
	14	1v2 3v5 4v6
	15	1v2 3v4 5v6

- 低的轮次将会比高的轮次先完赛，因此高轮次的不会发生重赛条件仅考虑低轮次中已进行的比赛，例如第二轮仅考虑第一轮，第三轮要考虑一二轮，依此类推。
- 每完成一轮比赛，每个队伍将改变其种子排名。A队的种子排名高于B队，当且仅当：1.A队的胜场大于B队 2.A队的胜场等于B队，A队已经对战过的手的胜场和减去负场和大于B队已经对战过的手的胜场和减去负场和 3.A队的胜场等于B队，A队已经对战过的手的胜

场和减去负场和等于B队已经对战过的对手的胜场和减去负场和，A队的初始种子排名高于B队。

然而 *RalZarek* 没有耐心读以上的规则说明，因此他希望你能帮他写一个程序，直接告诉他每个队伍的最终赛果。

Input

输入共有 15 行，第 i 行有 $16-i$ 个数，表示 $R_{i,j}$ ($1 \leq i \leq 15, 1 \leq j \leq 16-i$)， $R_{i,j} = 1$ 代表初始第 i 种子顺位的队伍与初始第 $i+j$ 种子顺位的队伍对战时初始第 i 种子顺位的队伍将获胜， $R_{i,j} = 0$ 代表初始第 i 种子顺位的队伍与初始第 $i+j$ 种子顺位的队伍对战时初始第 $i+j$ 种子顺位的队伍将获胜。

Output

输出共有 16 行，每行两个整数，第 i 行的第一个数表示初始第 i 种子顺位的队伍的胜场数，第 i 行的第二个数表示初始第 i 种子顺位的队伍的负场数。

Example

standard input	standard output
1 1 1 0 1 1 1 0 1 1 1 0 1 1 1	1 3
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	3 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1	3 0
1 1 1 0 1 1 1 1 1 1 1 1	3 1
1 1 1 0 1 1 1 0 1 1 1	3 2
0 1 1 1 1 0 1 1 1 1	3 2
1 1 1 1 1 1 0 1 1	2 3
1 1 1 0 1 1 1 1	2 3
1 1 1 1 1 1 1	3 1
1 1 1 0 1 1	1 3
0 1 0 1 1	1 3
1 1 1 1	3 1
1 1 1	2 3
1 1	3 2
1	0 3
	0 3

Note

样例解释：

Problem G. 神圣愤怒

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

RalZarek 最近在炉石中学习如何玩元气骑。这套牌的关键点在于将一张 125 费的卡牌置于牌库顶，并使用神圣愤怒抽取此卡造成 125 点伤害。然而，将这张卡恰好置于牌库顶是一件难事。某天，RalZarek 在观看杀戮尖塔直播时突然学到了全知之眼（看穿牌库），这使得控制牌序变为一件可以完成的事情。现在，RalZarek 组建好了他的卡组，并在对局开始时看穿了卡组中卡的顺序，他希望你能通过以下操作帮助他将这张关键卡置于牌库顶。

RalZarek 的卡组中有 n ($1 \leq n \leq 5000$) 张卡，由两种卡组成：关键卡与抽牌卡，均用一个数字 a_i ($0 \leq a_i \leq 5000$) 表示。其中 $a_i = 0$ 表示 125 费的关键卡， $a_i \neq 0$ 表示抽牌卡， a_i 为使用此卡时能抽取卡牌的数量。

初始时手牌为空。游戏开始后，可以进行以下三种操作：

1. 若牌库顶的卡牌**不是**关键卡，抽取牌库顶的一张牌，将其置于手中最右侧，否则无法从牌库顶抽取卡牌。
2. 若手中**最左侧**的卡牌点数为 k ，且牌库顶的 k 张卡牌**不是**关键卡，则使用手中最左侧的卡牌，并从牌库顶依次取 k 张牌放到手中最右侧，否则无法使用手中的卡牌。（已使用的卡牌将会消失）
3. 将牌库底的一张卡牌插入到牌库中的任意位置。

由于 RalZarek 的对手也可能使用了相同的卡组，因此他希望尽可能快速地将关键卡置于牌库顶。请你编写一个程序，计算将关键卡置于牌库顶需要的最小操作数。

Input

输入的第一行为一个整数 n ($1 \leq n \leq 5000$)，代表卡组中卡的张数。

输入的第二行为 n 个整数 a_i ($0 \leq a_i \leq 5000$)，代表卡组中的卡牌及其顺序，先输入的卡牌在后输入的卡牌的上方。

数据保证卡组中有且仅有一张关键卡，即有且仅有一个 $a_i = 0$ 。

Output

输出为一行一个整数，代表最小的操作数。

Examples

standard input	standard output
1 0	0
10 1 1 1 1 1 0 1 1 1 5	3
6 2 1 1 0 1 1	2

Note

样例一中，牌库里只有一个 0 且在牌库顶，不需要任何操作。

样例二中，先使用操作三将最底端的 5 移至顶端，再使用操作一抽到 5，再使用操作二打出 5，此时 0 在顶端，共 3 次操作。

样例三中，先使用操作一抽到 2，再使用操作二打出 2，此时 0 在顶端，共 2 次操作。
可以证明，不存在其他方法能使用更少的操作将 0 移至顶端。

Problem H. 最大节点和

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

现有一颗高度为 n 的满二叉树，共有 $2^n - 1$ 个节点。它的叶子节点的值是 0 或 1，其他节点的值是它两个子节点的异或值，记这颗树的所有的节点和为 s 。在所有满足这样要求的树中，最大的节点和 s 是多少？

对于一棵只有一个根节点的树的高度为 1。对于其他树，其高度为其子节点的最大高度加一。

由于答案可能很大，答案对 998244353 取模。

Input

每个测试点包含多个测试用例，第一行输入一个正整数 T ($1 \leq T \leq 10^5$)，表示测试用例的数量。接下来的 T 行中每行输入一个正整数数 n ($1 \leq n \leq 10^{18}$)，表示树的高度。

Output

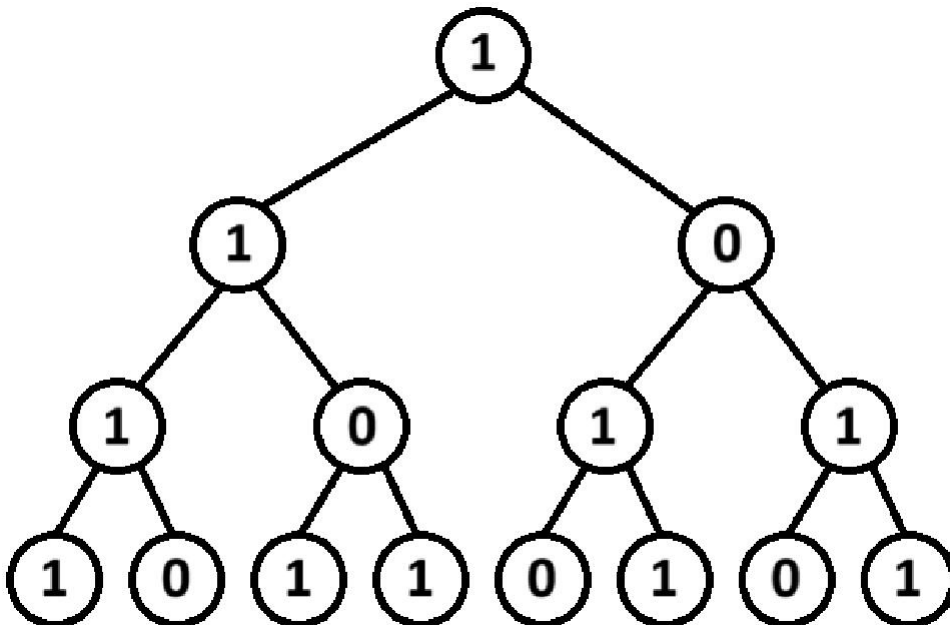
对于每个测试用例，输出一行一个正整数数 s ，表示最大的节点和。

Example

standard input	standard output
2	10
4	21
5	

Note

当 $n = 4$ 时，一个满足题意的树如下：



Problem I. 嘟咕噜之王

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 512 megabytes

嘟咕噜的个体值为 $[0, n]$ 范围内的整数。市场中所有个体值的嘟咕噜都有无限供应，其中个体值为 i 的嘟咕噜售价为 a_i 。

嘟咕噜之王 Rondo 初始拥有一只个体值为 0 的嘟咕噜。他可以进行以下操作：

1. 购买：在市场上购买任意个体值的嘟咕噜（花费对应售价）。
2. 融合：将购买的嘟咕噜与自己的嘟咕噜进行融合。融合后：
 - 被购买的嘟咕噜会被消耗（消失）。
 - Rondo 的嘟咕噜的个体值会更新为这两只嘟咕噜个体值的最小值和最大值之间的随机整数（在闭区间内均匀随机分布）。

Rondo 想要知道让他的嘟咕噜个体值达到 n 的最小期望花费。

Input

一行一个整数 n ($0 \leq n \leq 5000$)，表示嘟咕噜的个体值的最大值。

接下来一行 $n + 1$ 个整数，第 i 个整数表示个体值为 $i - 1$ 的嘟咕噜的售价 a_{i-1} ($1 \leq a_{i-1} \leq 10^9$)

Output

一行一个实数，表示最小期望花费。

答案被判定为正确当且仅当：输出结果与标准答案的绝对误差或相对误差不超过 10^{-6} 。（即若标准答案为 a ，你的输出为 b ，则需满足 $\frac{|a - b|}{\max(1, |a|)} \leq 10^{-6}$ ）

Example

standard input	standard output
2 10 24 100	248.0000000000

Problem J. 莲的简单题

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

可爱的莲希望给选手们舒适的比赛体验，于是出了这道题。



莲希望你构造一个长度为 n 的正整数数组 a ，其中的每个元素不超过 10^9 ，使得对于任意的下标 $i, j (i \neq j)$ ， $GCD(a_i, a_j) = GCD(i, j)$ ，且 $a_i \neq i$ 。其中 $GCD(x, y)$ 表示 x 和 y 的最大公约数。

Input

输入包含一个正整数 $n (2 \leq n \leq 1000)$ ，表示要构造的数组长度。

Output

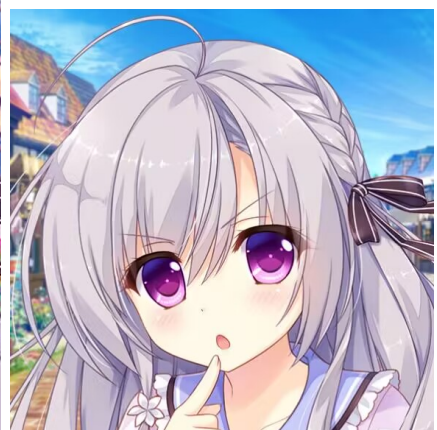
输出 n 个正整数表示数组 a ，两两之间以一个空格分隔

Example

standard input	standard output
4	5 14 33 52

Note

写题写累了不妨来看看莲吧，毕竟莲这么可爱，她会摸摸你的头跟你说加油的



Problem K. Capoo's stack

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

有 n 只可爱的 Capoo 想玩叠叠乐，每只 Capoo 都有一个体力值 a_i 。

为了防止在叠叠乐的过程中 Capoo 受到伤害，Capoo 们相互约定，除了在最上面的 Capoo，每只 Capoo 的体力值要大于等于它上面的所有 Capoo 的体力值之和。

Capoo 们自然希望参与叠叠乐的 Capoo 尽量多，所以它们向你提出了一个问题，在合理安排 Capoo 顺序的情况下，参与叠叠乐的 Capoo 数量最多为多少？

Input

本题包含多组测试数据。

输入第一行一个整数 T ，表示测试数据组数。($1 \leq T \leq 2 \times 10^5$)

对于每组数据，第一行一个整数 n ，表示 Capoo 数量。($1 \leq n \leq 2 \times 10^5$)

第二行 n 个整数 a_i ，第 i 个整数表示第 i 只 Capoo 的体力值。($1 \leq a_i \leq 10^9$)

数据保证 $\sum n$ 之和不大于 2×10^5 。

Output

对于每组测试数据，输出一行一个整数表示答案，每两组数据之间的答案需换行。

Example

standard input	standard output
2	4
5	2
1 3 4 7 9	
3	
1 1 1	

Note

对于**样例 #1**:

对于第一组测试数据，最优的一种方法是(Capoo顺序从低到高):

9 4 3 1

对于第二组测试数据，最优的一种方法是(Capoo顺序从低到高):

1 1

Problem L. 等价交换

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

在游戏《Minecraft》的知名模组"等价交换"中存在这样一个设定，每个物品存在一个 *EMC*(*Energy Matter Covalence*)，玩家可以将不需要的物品转化为对应的 *EMC*，并使用 *EMC* 换取需要的物品，从而降低游戏难度。

而作为一个比较懒的玩家，*agKc* 在游玩一些含有该模组的整合包时，会偷偷地给一些物品添加上 *EMC*。一个物品具有 *EMC* 的情况如下：

- 该物品初始具有 *EMC* 或玩家为其添加 *EMC*
- 合成该物品的所有物品均具有 *EMC*

一条合成表可表示为 $a_1 + a_2 + \dots + a_i \rightarrow b$ ，即可以用物品 $a_1, a_2, \dots, a_i$ 来合成物品 b ，同时，如果物品 $a_1, a_2, \dots, a_i$ 均具有 *EMC* 则物品 b 也具有 *EMC*。

现在 *agKc* 准备开坑一个新的含有"等价交换"模组的整合包，并提前查看了该整合包中的所有合成关系。该整合包中总共有 n 种物品与 m 条合成表，奇怪的是，所有的物品初始都不具有 *EMC*，所以他打算通过给一些物品加上 *EMC* 来使所有物品都具有 *EMC*。

由于 *agKc* 实在是太懒了，他把这个问题交给了你，要求你告诉他最少该为几件物品添加 *EMC* 才能使所有物品都具有 *EMC*。但 *agKc* 忘记告诉你合成表，所以你只好自己假设最佳情况了。

Input

第一行两个整数 n 和 m ($1 \leq n \leq 2 \times 10^5$, $0 \leq m \leq n$) 分别表示物品种数和合成表条数。

Output

输出一行一个整数，表示在最佳情况下 *agKc* 最少该添加 *EMC* 的物品的数量。

Example

standard input	standard output
3 2	1

Problem M. Maximize the total of maximum

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

坎格鲁斯普雷给了你一个长度为 n 的序列 a_i 和一个常数 c 。

定义一个非空子数组 $[l, r]$ 的权值为 $\max\{\sum_{i=l}^r a_i, c\}$ ，定义一个序列的权值为其所有非空子数组的权值之和。

现在，你可以进行至多 k 次操作，每次操作可以使一个序列中的元素 $+1$ 。

现在，坎格鲁斯普雷想问问你，在你进行完操作后，序列的权值最大值是多少？

Input

本题包含多组测试数据。

输入第一行一个整数 T ，表示测试数据组数。 $(1 \leq T \leq 2 \times 10^5)$

对于每组测试数据，输入第一行三个整数 n, k, c 。 $(1 \leq n \leq 2 \times 10^5, 0 \leq k \leq 10^9, -10^9 \leq c \leq 10^9)$

第二行 n 个整数 a_i ，表示题目中所给序列。 $(-10^9 \leq a_i \leq 10^9)$

数据保证 $\sum n$ 不大于 2×10^5 。

Output

对于每组测试数据，输出一行一个整数，表示答案。

Example

standard input	standard output
2	23
5 3 0	-13
1 2 -8 1 3	
4 1 -2	
-1 -1 -1 -1	

Note

对于样例 #1:

对于第一组测试数据:

其中一种最优解是 1 3 -8 3 3，该序列的权值计算过程如下:

$[1, 1] \sim [5, 5]: 1 + 3 + 0 + 3 + 3 = 10$

$[1, 2] \sim [4, 5]: 4 + 0 + 0 + 6 = 10$

$[1, 3] \sim [3, 5]: 0 + 0 + 0 = 0$

$[1, 4] \sim [2, 5]: 0 + 1 = 1$

$[1, 5]: 2$

总和为 $10 + 10 + 0 + 1 + 2 = 23$ 。

对于第二组测试数据:

最优解为 -1 0 -1 -1 或 -1 -1 0 -1。